

Amendments to the Claims:

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

1-13. (Canceled)

14. (Currently amended): A rendering method, comprising:

receiving at a rendering service a rendering request from a user site, the user site being in communication with the rendering service over a network, the rendering request comprising identifiers of rendering resources currently available at the user site required for performing a rendering task, wherein the rendering task is performed at the rendering service; maintaining at the rendering service a resource pool comprising rendering resources uploaded from the user site and rendering resources generated at the rendering service;~~from at least one previous rendering request from the user site;~~ comparing identifiers of the rendering resources in the resource pool at the rendering service with the identifiers of rendering resources currently available at the user site; selectively uploading rendering resources from the user site to the rendering service based on a result of said comparing step; and storing the selectively uploaded generated-rendering resources in the resource pool for use in processing additional rendering requests received from the user site.~~corresponding to previous rendering requests in the resource pool; and determining whether to generate a given raw resource into a generated rendering resource based on a result of the comparing step.~~

15. (Original): A rendering method according to claim 14, further comprising uploading a given required resource from the user site to the rendering service only if the

1 comparing step determines there is not a match between the resource pool and the
2 user site for that required resource.

1 16. (Original): A rendering method according to claim 15, the rendering
2 resources being uploaded to the rendering service in a raw format, the method further
3 comprising:
4 at the rendering service, generating the raw rendering resources to produce
5 generated rendering resources; and
6 providing the generated rendering resources to a rendering engine.

1 17. (Previously presented): A method according to claim 14, the rendering
2 resources comprising scene description files, further comprising manipulating a modeling
3 application such that said scene description files comprise at least one static scene description
4 file and at least one dynamic scene description file.

18-25. (Canceled)

1 26. (Currently amended): A method for rendering comprising:
2 receiving at a rendering service a rendering request from a user site to render one
3 or more images, the rendering request including information representative of one or more
4 required rendering resources used for rendering the one or more images; and
5 performing at the rendering service a rendering task in accordance with the
6 rendering request to produce at least one of the one or more images including processing one or
7 more of the required rendering resources,
8 wherein if a required rendering resource is not already stored in a data store local
9 to the rendering server computer system, then uploading that required rendering resource from
10 the user site,
11 wherein if a required rendering resource is already stored in the local data store,
12 then obtaining that required rendering resource from the local data store.

27. (Previously presented): The method of claim 26 wherein the processing includes performing a generation operation on a first required rendering resource to produce a first generated rendering resource, storing the first generated rendering resource, and providing the first generated rendering resource to a rendering engine.

28. (Previously presented): The method of claim 27 wherein performing the generation operation is performed only if the first required rendering resource is not already stored in the local data store.

29. (Previously presented): The method of claim 26 wherein the processing includes producing a generated rendering resource from a first required rendering resource, wherein if the first required rendering resource has been uploaded from the user site during servicing of a previous rendering request, then obtaining a previously generated rendering resource from the local data store thereby producing the generated rendering resource, wherein if the first required rendering resource has not been uploaded from the user site during servicing of a previous rendering request, then performing the uploading to obtain the first required rendering resource, performing a generation operation on the first required resource to produce the generated rendering resource, and storing the generated rendering resource in the local data store.

30. (Previously presented): The method of claim 26 further comprising: updating a resource pool comprising information representative of rendering resources that have been uploaded from the user site when a required resource is uploaded from the user site; comparing information associated with the required rendering resource with the information in the resource pool to determine whether or not a required rendering resource is already stored in the local data store.

1 31. (Previously presented): The method of claim 26 wherein the rendering
2 server computer system and the user site are at different geographical locations, and the method
3 further comprises communicating with the user site over a communication network.

1 32. (Previously presented): The method of claim 31 wherein the
2 communication network is the Internet.

1 33. (Previously presented): The method of claim 26 wherein the rendering
2 server computer system and the user site are co-located, and the method further comprises
3 communicating with the user site over a local area network

1 34. (Previously presented): The method of claim 26 wherein the required
2 rendering resources are raw rendering resource files, the method further comprising:
3 receiving from the user site a session control file comprising identities of the raw
4 rendering resources file required for the rendering task;
5 receiving from the user site at least one resource generation control file
6 comprising associations among the raw rendering resource files and a plurality of generated
7 rendering resources corresponding thereto; and
8 for each raw rendering resource file, performing (i) forward-mapping that raw
9 rendering resource file onto a set V of dependent generated rendering resources using
10 information derived from the at least one resource generation control file, (ii) reverse-mapping
11 each member of the set V onto a set W of raw rendering resource files using information derived
12 from the at least one resource generation control file; and (iii) marking that raw rendered
13 resource file for generation if (a) it is not identified in the resource pool or (b) any of the raw
14 rendering resource files set W required uploading for the rendering task.

1 35. (Previously presented): The method of claim 26 wherein the rendering
2 resources comprise scene description files, the method further comprising manipulating a
3 modeling application such that the scene description files comprise at least one static scene
4 description file and at least one dynamic scene description file.

1 36. (Previously presented): The method of claim 26 wherein the rendering
2 resources comprise one or more of scene description files, shader files, texture files, or
3 procedural files.

1 37. (Currently amended): A rendering server system comprising:
2 a server device connected to a first communication network for communication
3 with a user site,
4 the server device configured to receive a rendering request from the user site to
5 render one or images, the rendering request including information representative of one or more
6 required rendering resources used for rendering the one or more images,
7 the server device further configured to service the rendering request to produce at
8 least one of the one or more images wherein the server device processes one or more of the
9 required rendering resources,
10 the server device further configured to ~~upload request~~ a required rendering
11 resource from the user site if the required rendering resource is not already stored in a data store
12 local to the server device and to ~~store the uploaded~~ upload the required rendering resource in
13 from the user site to the local data store,
14 the server device further configured to access a required rendering resource from
15 the local data store if the required rendering resource is already stored in the local data store.

1 38. (Previously presented): The system of claim 37 wherein the rendering
2 resources comprise one or more of scene description files, shader files, texture files, or
3 procedural files.

1 39. (Previously presented): The system of claim 37 further comprising a
2 resource pool that is accessible by the server device, the resource pool comprising identities of
3 one or more rendering resources that have been uploaded from the user site, the server device
4 further configure to determine whether to upload a required rendering resource based on
5 information contained in the resource pool.

1 40. (Previously presented): The system of claim 37 further comprising a
2 rendering engine, wherein the server device produces a generated rendering resource suitable for
3 processing by the rendering engine, the generated rendering resource being stored on the local
4 data store.

1 41. (Previously presented): The system of claim 40 wherein if a required
2 rendering resource is already stored in the local data store, then the server device accesses the
3 local data store to obtain a generated rendering resource that corresponds to that required
4 rendering resource.

1 42. (Previously presented): A computer program product to be executed on a
2 server computer system for carrying out a network based rendering service, comprising:
3 a computer-readable storage medium; and
4 computer program code stored on the computer-readable storage medium,
5 wherein the computer program code is executable by a data processor and is configured to:
6 control the data processor to communicate with a user site to receive a
7 rendering request to render one or images, the rendering request including information
8 representative of one or more required rendering resources used for rendering the one or
9 more images;
10 control the data processor to perform a rendering task in accordance with
11 the rendering request to produce at least one of the one or more images wherein one or
12 more of the required rendering resources are processed by the data processor;
13 control the data processor to upload a required rendering resource from the
14 user site if the required rendering resource is not already stored in the local data store and
15 to store the uploaded rendering resource in the local data store; and
16 control the data processor to access the local data store to obtain a required
17 rendering resource if the required rendering resource is already stored in the local data
18 store.

1 43. (Previously presented): The computer program product of claim 42
2 wherein the rendering resources comprise one or more of scene description files, shader files,
3 texture files, or procedural files.

1 44. (Previously presented): The computer program product of claim 42
2 wherein the computer program code is further configured to control the data processor to
3 maintain a resource pool comprising identities of one or more rendering resources that have been
4 uploaded from the user site.

1 45. (Previously presented): The computer program product of claim 42
2 wherein the computer program code is further configured to control the data processor to process
3 a required rendering resource to produce a generated rendering resource suitable for processing
4 by a rendering engine, and to store the generated rendering resource on the local data store.

1 46. (Previously presented): The system of claim 45 wherein if a required
2 rendering resource is already stored in the local data store, then a generated rendering resource
3 that corresponds to that required rendering resource is obtained from the local data store.